Improved Direct Volume Visualisation of the Coronary Arteries Using Fused Segmented Regions

Daniel Mueller
School of Engineering Systems
Faculty of Built Environment and Engineering
Queensland University of Technology (QUT)
Brisbane, QLD, Australia
d.mueller@qut.edu.au

Anthony Maeder
e-Health Research Centre
CSIRO ICT Centre
Brisbane, QLD, Australia
anthony.maeder@csiro.au

Peter O'Shea
School of Engineering Systems
Faculty of Built Environment and Engineering
Queensland University of Technology (QUT)
Brisbane, QLD, Australia
pj.oshea@qut.edu.au

Abstract

Coronary heart disease was the single largest cause of sudden death in Australia in 2002. Computed tomography angiography (CTA) provides high resolution, high contrast images of the thoracic cavity, and as such has emerged as the imaging modality of choice for diagnosing and planning treatment for coronary heart disease. However, radiologists and cardiac surgeons require tools to easily identify possible stenosis (narrowed or constricted coronary vessels) in such CTA datasets. We present a method which allows users to interactively visualise a specific three-dimensional region of interest (ROI). In our example, segmentation methods are applied to isolate the coronary vessels, which in turn are visually enhanced using various perceptual cues. The segmentation is achieved using a combination of thresholding, region-growing, and morphological operations. The perceptual enhancement is realized by fusing direct volume rendered images using weighting factors determined by the segmented regions. The user can allow for the easy dissemination of relevant information by adjusting 'transfer functions' to control the degree of ROI enhancement. This approach requires only roughly segmented regions of interest, and allows for the 3D visualisation of calcifications within vessels. This proposed method has significant potential for helping to facilitate the efficient treatment for coronary heart disease. Furthermore, it can be implemented at interactive framerates on comparatively cheap, desktop computing hardware making it readily accessible.

1. Introduction

Cardiovascular disease (CVD) is the leading cause of death in a number of different countries. More specifically, according to a recent study [20], coronary heart disease was the single largest cause of sudden death in Australia in 2002. This disease stems from the underlying problem of *atherosclerosis*, which is a build up of plaque (consisting of substances including, among others, cholesterol and calcium) on the interior surface of arteries supplying the heart. Coronary heart disease typically manifests in two forms: heart attack, and angina. A heart attack occurs when blood flow is completely blocked, typically from a dislodged portion of plaque. Angina, typically brought on by physical activity, is a chest pain or discomfort caused by an inadequate blood flow due to the narrowed artery.

Computed tomography angiography (CTA) has emerged as the imaging modality of choice for diagnosing and planning treatment for coronary heart disease. An intravenous contrast agent (such as iodine-based dye or another substance with high molecular weight) is used to enhance the visibility of blood, and hence the carrier vessels [14]. The areas containing the *in vivo* contrast agent are marked in the resultant output images with a large Hounsfield unit (HU).[†]

Radiologists and cardiac surgeons require tools to help identify and visualise stenosis within the coronary arteries. Current medical imaging workstations include a number of two-dimensional (2D) tools such as multiplanar reformatting (MPR), oblique sectioning, and maximum intensity projection (MIP) [12]. To help manage the three-dimensional (3D) information, state of the art workstations are now beginning to include surface rendering algorithms. Figure 1 depicts examples of multi-slice MIP and surface rendering. Unfortunately, surface rendering approaches (whereby an explicit surface is extracted and converted to polygons by the Marching Cubes algorithm [7]) typically suffer from the problem of information occlusion, in which *external* surfaces obstruct *internal* surfaces. While solving the issue of information occlusion, traditional direct volume rendering (whereby surfaces of interest are interactively classified using transfer functions) can suffer from

[†]The Hounsfield unit is the arbitrary greyscale unit used by CT scanners.

Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA 2005)
0-7695-2467-2/05 $20.00 © 2005 IEEE

the opposite problem of information overload. Information overload occurs when too many input pixels are mapped to a single output pixel, typically resulting in blurry images [4]. To appropriately visualise the coronary arteries, and to overcome both the problems of information occlusion and overload, we present an approach which fuses direct volume rendered images with weights calculated using regions of interest (ROIs).

In the proposed approach, the heart and vessels of interest are segmented using a combination of thresholding, region-growing, and morphological filtering. Traditional classification methods (using transfer functions [6]) are applied to each region of interest, which are lastly arithmetically fused to form the final rendition. The direct volume rendering is performed using 3D texture-mapping on graphics accelerated hardware, and the fusion is accomplished by linearly compositing the two regions using a fragment shader program. Early results indicate this approach has significant potential for helping facilitate diagnosis and treatment for coronary heart disease.

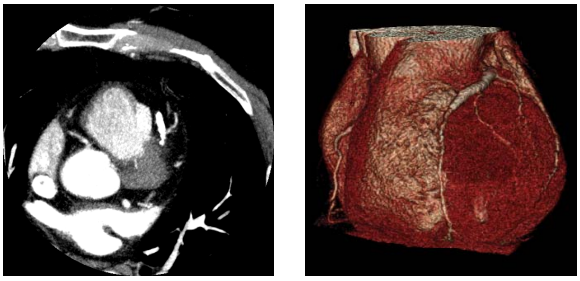

Figure 1. Left: Axial eight-slice MIP of coronary CTA dataset. Right: Surface rendering available on state of the art medical imaging workstations.

2. Materials and Method

The following sections detail the specifics for preparing the data, segmenting the vessel and heart regions, and finally fusing and rendering the data. The Insight Toolkit (ITK) [19] was extensively used to perform operations in the data preparation and segmentation phases. It should also be noted that all operations for the proposed method are applied in true 3D space.

2.1. Data Preparation

Computed tomography angiography (CTA) datasets can be obtained of the thoracic cavity and typically include structures such as the heart, lungs, rib cage, and liver. For the purposes of this study, three datasets were acquired using a Phillips Brilliance 40 CT scanner *(Phillips, MA, USA)*. The data was extracted from the relevant DICOM series as signed 16-bit volumes (note that while 16-bits were allocated, only 12-bits were required to capture the $[-2000, +2000]$ HU range).

Following this, a noise-reducing curvature anisotropic diffusion filter was applied. This filter is a non-linear, edge-preserving filter which is based around a modified curvature diffusion equation (MCDE) [5]. It is well suited to our purposes because it is does not suffer from edge-enhancing properties associated with traditional anisotropic diffusion, and it also tends to preserve finer detailed structures (such as vessels). The `itk::CurvatureAnisotropicDiffusionImageFilter` was applied to the datasets with iterations=3, time step=0.0625, and conductance=5.0.

Next the data was quantised to 8-bits. This was necessary because the rendering software currently only supports 8-bit data. The `itk::RescaleIntensityImageFilter` was applied to rescale the data to have the new range of $[0, 255]$. To help manage the size of the data, the volumes were also scaled by a factor of 2 in the x-y plane. For example, Dataset #1 was scaled from $512 \times 512 \times 428$ pixels to $256 \times 256 \times 428$ pixels, with a final physical resolution of 0.7 mm $\times 0.7$ mm $\times 0.4$ mm. The `itk::ResampleImageFilter` was used to perform the scaling.

Finally, to focus our interest to a representative range of values, a window/level intensity mapping was applied to the data. This mapping is a common operation in medical imaging whereby the user interactively specifies the width of the window, and the center of the window (known as the level). All values below and above the window are mapped to the datatype min and max respectively, and the values within the window are linearly rescaled. The `itk::IntensityWindowingImageFilter` was applied to all datasets with the typical values of window=80 and level=125. The data preparation phase is depicted in Figure 2.

2.2. Segmentation

As discussed further in Section 2.3, the proposed approach allows for pre-segmented regions of interest to be visually enhanced. In this particular case, we are interested in segmenting the vessels and heart. One of the strengths of the proposed rendering technique is that only roughly segmented ROIs are required because further segmentation (i.e. classification) will be applied during the rendering stage. The following sub-sections describe the segmentation algorithms used to create the approximate binary representations of the vessels and heart.

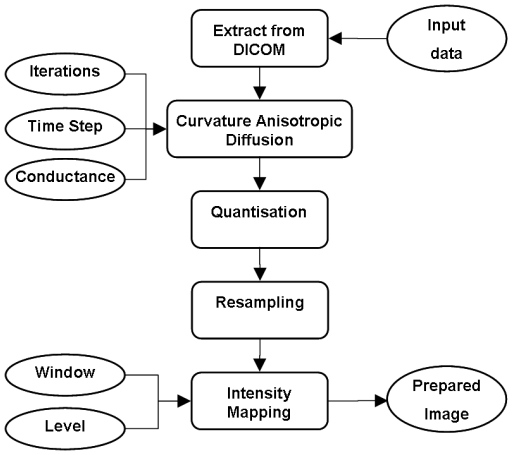

Figure 2. Collaboration diagram for data preparation phase.

2.2.1 Vessels

A number of different vessel segmentation methods have been reported in current literature. Vessel *tracking* approaches generate a central axis directly from the dataset, and are typically iterative in nature [17, 16]. On the other hand, *pattern-recognition* methods segment the vessels based on some form of image processing such as thresholding or region-growing [2, 1, 18], and then use a thinning algorithm to calculate the centreline (if required) [8]. The proposed method can be classified as a pattern-recognition approach because it combines morphological filtering (to remove large, non-target structures) and region-growing (to further segment the finer vessel-like structures).

The vessel segmentation task starts by morphologically removing large objects which are not part of the vessels. This is accomplished by applying a greyscale erosion operation to remove objects 'smaller' than the structuring element. A greyscale dilation operation is then applied to return the remaining 'large' objects to their original size. Using morphology terminology, an erosion followed by a dilation is known as an 'opening'. Following this, the opened image is subtracted from the original, creating an intermediate image containing only small objects (such as vessels). The `itk::GrayscaleErodeImageFilter` and `itk::GrayscaleDilateImageFilter` were used with the structuring element `itk::BinaryBallStructuringElement`. The `itk::SubtractImageFilter` was used to performed the image subtraction between the output of the data preparation and the output of the dilation filter. It was found that a structuring element with radius=4 captures the desired vessels. Figure 3 depicts this morphological filtering process.

Next, a window/level intensity mapping was applied to the subtracted image. This mapping is required to help delineate the vessels from ambient noise. Values of window=80 and level=60 ensure that the high-intensity vessels are preserved.

The final task of the vessel segmentation process was a region-growing operation to create the desired binary volume. Region-growing is a versatile neighbourhood method, requiring a starting *seed* point. In a recursive fashion, starting with the seed, each neighbour of the current pixel is visited and added to the region if it satisfies a simple intensity threshold [3]. In our case, the user specifies a single seed point in the vessel of interest (eg. the right or left coronary artery). From this seed point, all connected pixels (within the pre-defined threshold) are included in the final output volume. It should be noted that traditional region-growing approaches can be adversely affected by vessels with irregularly varying intensities and/or vessels connected to large structures with similar intensities [17]. However, the proposed approach helps alleviate such problems through the application of morphological and window/level filtering. The actual region-growing was performed by an `itk::ConnectedThresholdImageFilter` with lower threshold=65 and upper threshold=255. The components performing the vessel segmentation are depicted in Figure 4.

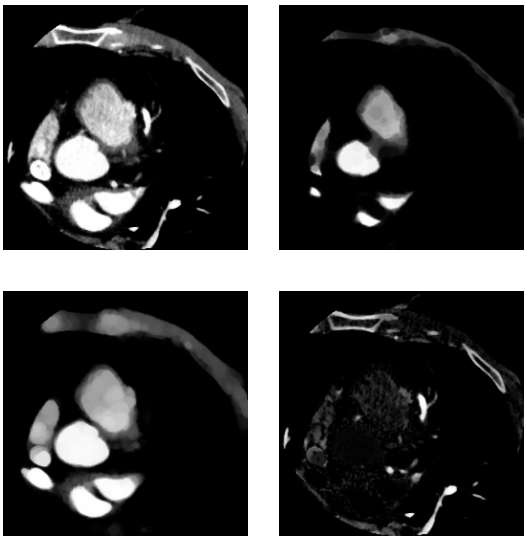

Figure 3. Top-Left: Original axial slice of prepared data. Top-Right: Eroded image by structuring element with radius=4. Bottom-Left: Dilated image by structuring element with radius=4. Bottom-Right: Dilated image subtracted from prepared image.

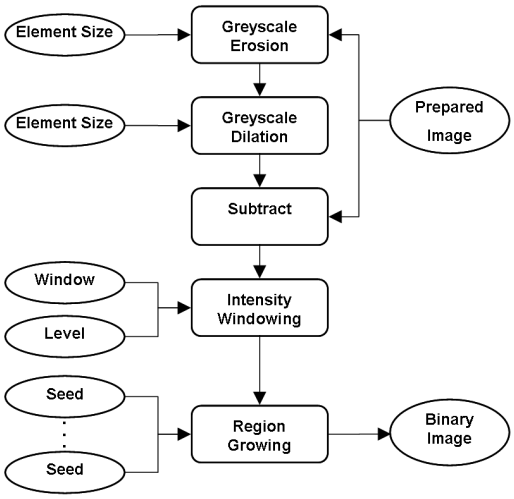

Figure 4. Collaboration diagram for vessel segmentation phase.

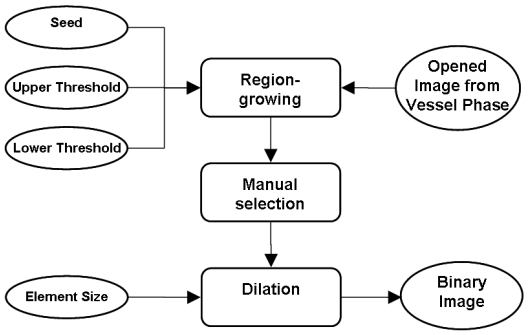

Figure 5. Collaboration diagram for heart segmentation phase.

2.2.2 Heart

In order to clearly distinguish the heart tissue, it must be segmented from other tissue in the CTA data including the rib cage, liver, and lungs. For our purposes a region-growing approach was adopted. Unfortunately region-growing can be adversely affected by noisy images, such as ours. However, because we are only interested in the large anatomical structures (ie. heart wall, chambers, major arteries, etc.), we can use the greyscaled opened intermediate image from the previous vessel segmentation phase. This image has already been processed to contain only large structures (with the small structures eroded by the structing element).

The `itk::ConnectedThresholdImageFilter` was applied to all datasets with a user specified seed anywhere within the heart or major arteries. The user interactively specifies the lower threshold until the rib cage and liver are excluded from the region. Finally, the region is morphologically dilated with an `itk::BinaryBallStructuringElement` of radius=1 to ensure the heart boundaries are included the output binary image. Figure 5 depicts the algorithm used to segment the heart region.

2.3. Fusing and Rendering

Direct volume rendering can be implemented using 3D texture-mapping with programmable fragment shaders [6, 11]. This approach uses OpenGL (Open Graphics Library) and compatible accelerated graphics hardware. The textures are simply volumes comprised of four 8-bit channels, namely RGBA (red, green, blue, alpha). We used all four channels: *alpha* to contain the CTA value data, *red* to contain the CTA gradient magnitude data, *green* to contain the heart region, and *blue* to contain the vessel region.

At the application level, the rendering system slices the RGBA 3D texture with view-aligned polygons. These polygons, sometimes referred to as 'proxy-geometry', trilinearly interpolate the texture from the given viewing angle. The final image is produced by compositing each of the view-aligned images in back-to-front order. Compositing is often called the 'over' operator, and is implemented as:

$$\begin{aligned} c_f &= \sum_{i=0}^{n-1} c_i \times \prod_{j=0}^{i-1} 1-\alpha_j \qquad (1)\\ &= c_0 + c_1(1-\alpha_0) + c_2(1-\alpha_0)(1-\alpha_1) + \ldots \\ &= c_0 \; over \; c_1 \; over \; c_2 \; over \; \ldots \; over \; c_{n-1} \end{aligned}$$

where α_i is the opacity at pixel i, c_i is the pre-multiplied (R, G, B) colour vector at pixel i, and c_f is the final composited colour. Within the rendering pipeline, our custom fragment shader program is executed for each interpolated (x, y, z) sample from the texture. It is within the fragment shader that we implement classification and fusion [9].

Transfer functions are used to classify the volume as different materials. In direct volume rendering, it is common to associate multiple scalar values with each pixel [6, 13, 15]. We have chosen to apply a two-dimensional classification domain, which uses the CTA value (f) and first derivative magnitude (∇f) to determine the material type of each pixel. The transfer function is implemented as a lookup table (LUT) using a 2D texture. The fragment shader uses the interpolated pixel values (f' and $\nabla' f$) as indices into the classification LUT. The current pixel (known as a 'fragment') is assigned the resultant RGBA value from the table lookup, classifying the location as a specific material. The gradient magnitude (∇f) volume is calculated using the `itk::GradientMagnitudeImageFilter`.

Numerous types of pixel-level fusion have been identified in current literature; including arithmetic, statistical, fuzzy, and multi-resolution techniques [10]. For two images, arithmetic fusion can be defined as:

$$F(i,j,k) = w_1 I_1(i,j,k) + w_2 I_2(i,j,k) + C \qquad (2)$$

where $F(i,j,k)$ is the resultant pixel in the fused image, w_1 is a constant contribution for image $I_1(i,j,k)$, w_2 is a constant contribution for image $I_2(i,j,k)$, and C is a constant mean offset. The simplicity of arithmetic fusion means it is applicable to either scalar (i.e. greyscale) or vector (RGB, RGBA) images. In our particular case we are fusing *after* classification, and therefore are dealing with vector (RGBA) images. In line with the concept of compositing, we assign the weights as follows:

$$\begin{aligned} w_1 &= h(i,j,k) \times [1 - v(i,j,k)] \\ w_2 &= v(i,j,k) \\ C &= 0 \end{aligned}$$

where $h(i,j,k)$ is the interpolated heart region pixel, and $v(i,j,k)$ is the interpolated vessel region pixel. The choice of these values result in only the heart and vessel regions being rendered, with the vessels appearing *over* the heart.

The data flow within the fragment shader is depicted in Figure 6. Listing 1 shows the fragment shader program used to implement the classification and fusion.

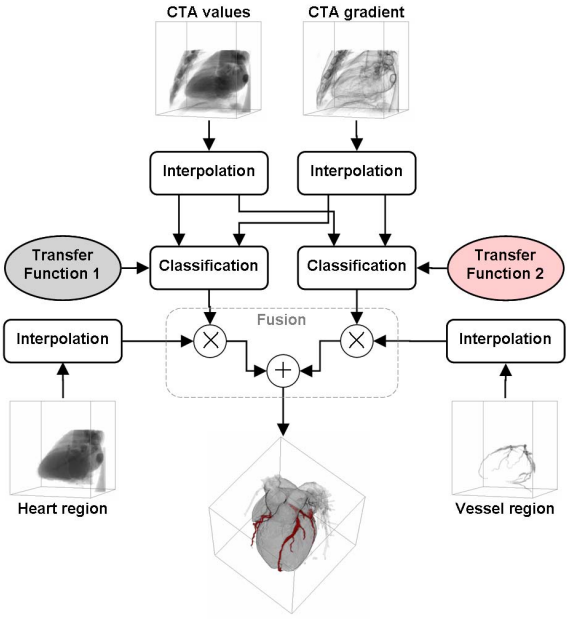

Figure 6. Data flow diagram for the rendering phase.

Listing 1. Fragment shader program used to perform 2-D classification and fusion (OpenGL Shading Language)

```
//Texture samplers
uniform sampler3D sampler_texture0;
uniform sampler2D sampler_tf_1;
uniform sampler2D sampler_tf_2;

//Texture coords
varying vec3 texture0_coord;

void main()
{
  //Interpolate 3D texture
  vec4 tex_0 = texture3D(sampler_texture0,
                         vec3(texture0_coord));

  //NOTE:
  //  A = Value
  //  R = Gradient
  //  G = Region 1
  //  B = Region 2

  //Classify using 2D transfer functions
  vec2 tf_coords = vec2(tex_0.r, tex_0.a);
  vec4 tf_1 = texture2D(sampler_tf_1, tf_coords);
  vec4 tf_2 = texture2D(sampler_tf_2, tf_coords);

  //Fuse
  vec4 final_rgba = vec4(0.0,0.0,0.0,0.0);

  final_rgba = vec4(mix(final_rgba,
                        tf_1.rgba,
                        tex_0.g));

  final_rgba = vec4(mix(final_rgba,
                        tf_2.rgba,
                        tex_0.b));

  //Set final color
  gl_FragColor = final_rgba;
}
```

3. Results and Discussion

The fusion and rendering was implemented on an ATI Radeon 9800 Pro graphics processing unit (with 128MB of VRAM) using OpenGL. All portions of the application were executed on an Intel Pentium 4, 2.8GHz processor, 1GB RAM. The rendering system performed at $\approx$ 2fps (frames per second). However, the introduction of an interactive mode (whereby the rendition quality is reduced during user interaction) allowed the system to increase performance to near real-time framerates ($\approx$ 10fps) during interaction.

Rather than comparing the vessel segmentation algorithm with an expert, hand- traced 'gold-standard', visual comparison was used to evaluate the effectiveness of the proposed algorithm. To facilitate this, the prepared CTA data was arithmetically fused with the vessel region, to produce a simple image average (i.e. $w_1 = 0.5$, $w_2 = 0.5$, and $C = 0$). This approach allows for the easy identification of false-positives (i.e. objects incorrectly included in the vessel region). A number of these anomalies were identifed

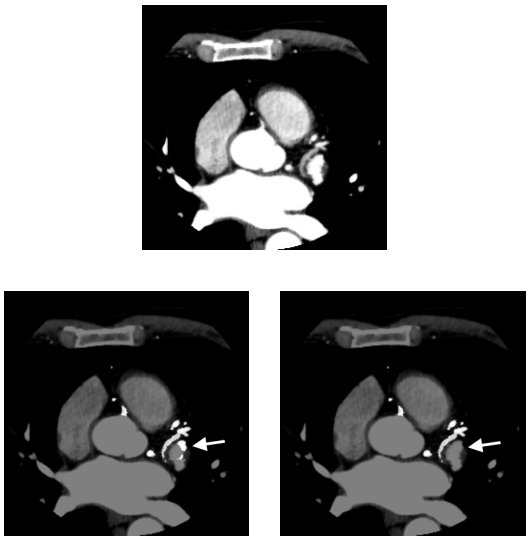

Figure 7. False-positives within the segmented vessel region can be visually identified in the average image (bottom-left). Top: Axial slice from prepared data. Bottom-Left: Average of prepared and segmented vessel data. Bottom-Right: Average of prepared and edited segmented vessel data (used for rendering).

in the datasets, with each occurring in structures adjacent to vessels with similar morphometric properties (see Figure 7). The small number of false-positives were manually removed before rendering.

Resultant images validate the assertion that the proposed method solves the problem of both information occlusion and overload. Figure 8 depicts a dataset rendered with surface-like appearance. Notice that the coronary arteries are perceptually conspicuous, and (in the bottom sub-figure) the volume can be rendered semi-transparently preventing information occlusion. The algorithm can also be used to augment existing techniques. Figure 9 shows a series of renderings similar to maximum intensity projection (MIP), however, unlike classical MIP the vessels can be clearly distinguished from the surrounding tissue.

One of the advantages of the proposed method is the two-step segmentation process - that is the segmentation of the regions of interest (in this case the heart and coronary vessels), followed by the classification. The transfer function classification scheme associated with direct volume rendering allows users the ability to *interactively* fine-tune the roughly segmented ROI. This intrinsic coupling of segmentation and visualisation enhances the flexibility of the approach.

Another advantage of the proposed method is the ability to apply 2D (value and gradient magnitude) classification within the segmented regions. This results in the ability to easily visualise and assess the severity of stenosis caused by calcification. Figure 10 shows a rendition clearly depicting the extent of calcification in the left coronary artery. These images demonstrate the potential for helping clinicians better visualise the 3D anatomical structures of the coronary vessels.

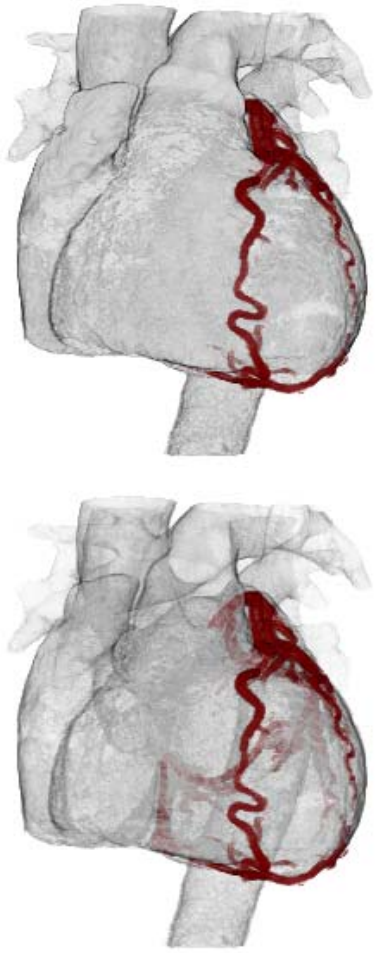

Figure 8. Dataset #1 rendered using the proposed method with surface-like appearance. Information occlusion can be eliminated by classifying obstructing tissue with semi-transparent properties. Note in the bottom figure the left coronary artery is visible from behind the pulmonary trunk.

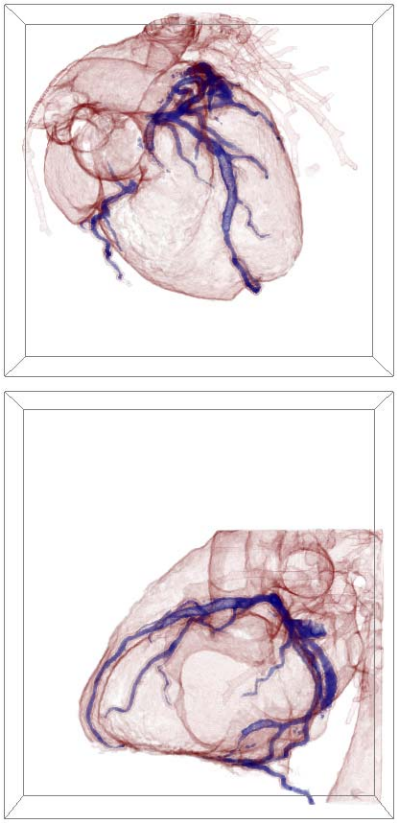

Figure 9. Dataset #2 rendered using the proposed method with appearance similar to maximum intensity projection (MIP). Note the vessels can be clearly distinguished from the surrounding tissue.

Figure 10. Dataset #3 rendered with surface-like appearance. Note the calcifications in the left coronary artery (classified as blue).

4. Conclusions and Future Work

A novel visualisation technique was presented allowing users to render volumes with visually enhanced segmented regions of interest. This method was applied to the problem of diagnosing and planning treatment of stenosis within coronary arteries. This approach allows for the interactive 3D visualisation of coronary arteries, including areas of calcification. Resultant images from this approach show the potential for helping facilitate treatment for coronary heart disease.

The method was divided into three distinct phases: data preparation, region segmentation, and fusion/rendering. The data was prepared for visualisation by applying a number filters performing smoothing, quantisation, and intensity transformation. The heart region was then segmented using a combination of morphological filtering and region-growing. The coronary vessels were segmented based on their morphometric properties. Vessel-shaped objects were extracted by subtracting a morphological greyscale opened image from the original image. Region-growing was then applied to extract the binary vessel image. These segmented regions were then used as weights for arithmetic fusion of independently classified volumes.

This paper presented a preliminary examination of our results. Future work will involve undertaking a perceptual study from a group of radiologists and surgeons to validate the approach. The segmentation method should also be further analysed and quantitatively validated. Finally, the use of deformable model segmentation (such as level sets) could be explored for the autonomous segmentation of the heart from the rib cage and liver.

Acknowledgements

Thanks are extended to Dr. Richard Slaughter of the Prince Charles Hospital for providing data and his clinical expertise.

References

[1] T. Boskamp, D. Rinck, F. Link, B. Kummerlen, G. Stamm, and P. Mildenberger. New vessel analysis tool for morphometric quantification and visualization of vessels in ct and mr imaging data sets. *Radiographics: a Review Publication Of The Radiological Society Of North America, Inc*, 24(1):287–297, 2004.

[2] Z. Chen and S. Molloi. Automatic 3d vascular tree construction in ct angiography. *Computerized Medical Imaging and Graphics*, 27(6):469–479, 2003.

[3] B. M. Dawant and A. P. Zijdenbos. Image segmentation. In M. Sonka and J. M. Fitzpatrick, editors, *Handbook of Medical Imaging: Medical Image Processing and Analysis*, volume 2, pages 71–127. SPIE, Bellingham, 2000.

[4] H. Hauser, L. Mroz, G.-I. Bischi, and M. Groller. Two-level volume rendering - fusing mip and dvr. In *Visualization*, pages 211–218. IEEE, 2000.

[5] L. Ibáñez, W. Schroeder, L. Ng, and J. Cates. The itk software guide. Version 2.0, 2005, Available online: www.itk.org/ItkSoftwareGuide.pdf.

[6] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[7] B. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *International Conference on Computer Graphics and Interactive Techniques*, pages 163 – 169. ACM, 1987.

[8] C. M. Ma and M. Sonka. A fully parallel 3d thinning algorithm and its applications. *Computer Vision and Image Understanding*, 64(3):420–433, 1996.

[9] D. Mueller, A. Maeder, and P. O'Shea. Implementing direct volume visualisation with spatial classification. In *Workshop on Digital Image Computing*, pages 49–54. APRS, 2005.

[10] V. Petrovic. *Multisensor pixel-level image fusion*. Phd dissertation, University of Manchester, 2001.

[11] C. Rezk-Salama. *Volume rendering techniques for general purpose graphics hardware*. Phd dissertation, University Erlangen-Nuremberg, 2001.

[12] R. A. Robb. Three-dimensional visualization in medicine and biology. In I. N. Bankman, editor, *Handbook of Medical Imaging : Processing and Analysis*, pages 685–712. Academic Press, San Diego, 2000.

[13] H.-o. Shin, B. King, M. Galanski, and H. K. Matthies. Use of 2d histograms for volume rendering of multidetector ct data: development of a graphical user interface. *Academic Radiology*, 11(5):544–550, 2004.

[14] M. Sonka, A. Stolpen, W. Liang, and R. M. Stefancik. Vascular imaging and analysis. In M. Sonka and J. M. Fitzpatrick, editors, *Handbook of Medical Imaging: Medical Image Processing and Analysis*, volume 2, pages 809–914. SPIE, Bellingham, 2000.

[15] N. Svakhine, D. Ebert, and D. Stredney. Illustration motifs for effective medical volume illustration. *IEEE Computer Graphics and Applications*, 25(3):31–39, 2005.

[16] S. Wesarg and E. A. Firle. Segmentation of vessels: the corkscrew algorithm. In *Medical Imaging: Image Processing*, volume 5370, pages 1609–1620. SPIE, 2004.

[17] O. Wink, W. Niessen, and M. Viergever. Fast delineation and visualization of vessels in 3-d angiographic images. *IEEE Transactions on Medical Imaging*, 19(4):337–346, 2000.

[18] Y. Yang, A. Tannenbaum, D. Giddens, and W. H. Coulter. Knowledge-based 3d segmentation and reconstruction of coronary arteries using ct images. In *26th Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*, volume 3, pages 1664 – 1666. IEEE, 2004.

[19] T. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxes, and R. Whitaker. Engineering and algorithm design for an image processing api: a technical report on itk - the insight toolkit. In *Medicine Meets Virtual Reality*, pages 586–592, 2002.

[20] Heart, stroke and vascular diseases - australian facts 2004. Technical Report CVD27, Australian Institute of Health and Welfare (AIHW), 2004, Available online: www.aihw.gov.au/publications/cvd/hsvd04/hsvd04.pdf.